Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 3– No.4, November 2015 – www.caeaccess.org

Is It True for Static Scaling Cloud Model there Exists a Centrally Asymmetric Static Workload Pattern?

Ravi (Ravinder) Prakash G.
Senior Professor Research,
BMS Institute of Technology
Dodaballapur Road, Avalahalli, Yelahanka,
Bangalore – 560 064

Kiran M.
Research Scholar
School of Computing & Information Technology
REVA University, Yelahanka,
Bengaluru - 560064

ABSTRACT

Actual quantifiability is a concept in static scaling that is based on two assumptions: (1) every cloud service provider is cautious, i.e., does not exclude any cloud consumer's static workload resource pooling pattern choice from consideration, and (2) every cloud service provider respects the cloud consumer's static workload resource pooling pattern preferences, i.e., deems one cloud consumer's static workload resource pooling pattern choice to be infinitely more likely than another whenever it premises the cloud consumer to prefer the one to the other. In this paper we provide a new approach for actual quantifiability, by assuming that cloud service providers have asymmetric static workload resource pooling pattern about the cloud consumer's static workload utilities. We show that, if the uncertainty of each cloud service provider about the cloud consumer's static workload utilities vanishes gradually in some regular manner, then the static workload resource pooling pattern choices it can quantifiably make under common conjecture in quantifiability are all actually quantifiable in the original static scaling with no uncertainty about the cloud consumer's utilities.

Keywords

Cloud service provider, cloud consumer, static workload, asymmetric, resource pooling pattern, utilities, static scaling, behavioral, quantifiability

1. INTRODUCTION

Static scaling deals with the ways the cloud service providers may reason about its cloud consumers before making a decision. More precisely, in Static scaling cloud service providers base its static workload resource pooling pattern choices on the conjectures about the cloud consumers' behavior, which in turn depend on its conjectures about the cloud consumers' conjectures about other cloud consumers' behavior, and so on [1] [7] [9] [21]. A major goal of Static scaling in this work is to study such conjecture hierarchies, to impose reasonable conditions on these, and to investigate its resource pooling pattern behavioral implications.

A central idea in Static scaling is common conjecture in quantifiability, stating that a cloud service provider premises that its cloud consumers choose quantifiably, and so on. In our view, one of its most natural refinements is the concept of actual quantifiability. Actual quantifiability is based on the following two conditions: The first states that cloud service providers are cautious [2] [8] [10] [22], meaning that they do not exclude any cloud consumers' static workload resource pooling pattern choice from consideration. The second condition states that whenever premise that a static workload resource pooling pattern choice a is better than another static workload resource pooling pattern choice b for a cloud consumer, then the probability assign to b must be at most α times the probability assign to a. Under α-actual quantifiability there is common conjecture in the event that every cloud service provider is cautious and satisfies the α-actual trembling condition. A static workload resource pooling pattern choice is called actually quantifiable if it can be chosen under α-actual quantifiability for every $\alpha > 0$ [3] [11] [15] [20].

2. RESEARCH CLARIFICATION

The usual interpretation of actual quantifiability assumes that cloud consumer makes mistakes, but that deem more costly mistakes much less likely than less costly mistakes. In this paper we offer a rather different approach for actual quantifiability. Instead of assuming premise cloud consumer to make mistakes, we rather suppose that have uncertainty about its utility function, while believing that it chooses quantifiably. We thus consider a static scaling with asymmetric static workload resource pooling pattern. Our result states that, if we let uncertainty about the cloud consumer's utility go to zero in some regular manner, then every static workload resource pooling pattern choice that can quantifiably be made under common conjecture in quantifiability in the static scaling with asymmetric static workload resource pooling pattern, will be actually quantifiable in the original static scaling, in which there is no uncertainty about the cloud consumer's utilities.

In the static scaling with asymmetric static workload resource pooling pattern, we impose some regularity conditions on the cloud service providers' conjectures about the cloud consumer's utility functions which can be summarized as follows: First, for every outcome in the static scaling, the conjecture that cloud service provider i has about cloud service provider j's utility from this outcome, is always normally distributed with its mean at the "original" utility in the original static scaling. As a consequence, cloud service provider i deems any utility function possible for cloud service provider j, and hence every resource pooling pattern choice for cloud service provider j can be optimal for some utility function deemed possible by i. Together with the condition that i premises in j's quantifiability, this actually makes sure that cloud service provider i deems every static workload resource pooling pattern choice possible for cloud service provider j, thus mimicking the cautiousness condition described above. Secondly, i's conjecture about j's utility function should be independent from its conjecture about j's conjecture hierarchy. This makes intuitive sense since j's conjecture hierarchy is analytic property of this cloud service provider, whereas its utility function is not analytic property [4] [12] [16] [23]. Therefore there is no obvious reason to expect any correlation between these two characteristics. Thirdly, i's conjecture about j's utilities from different outcomes in the static scaling should be independent from each other. Possibly some of these conditions can be relaxed for the proof of our result, but we leave this issue for future research.

The paper is organized as follows: In Section 3 we introduce our static scaling model [5] [13] [17] [24] for static scaling with asymmetric static workload resource pooling pattern, we formalize the idea of common conjecture in quantifiability for these static scaling, and show that common conjecture in quantifiability is always possible (Descriptive Study I). In Section 4 we introduce our static scaling model for static scaling with symmetric static workload resource pooling pattern, and present the concept of actual quantifiability for these static scaling (Prescriptive Study). In Section 5 we state our result, establishing the connection between common conjecture in quantifiability in the static scaling with asymmetric static workload resource pooling pattern in the presence of small uncertainty about the cloud consumer's utility function, and actual quantifiability in the original static scaling (Descriptive Study II). In Section 6 we provide some concluding remarks. All proofs are collected in Section 7.

3. DESCRIPTIVE STUDY I

3.1 Static Scaling Model

Throughout this paper we restrict attention to static scaling operations with two sets of cloud service provider. Let $\delta = (C_i, w_i)_{i\epsilon I}$ be a finite, static workload where $I = \{1, 2\}$ is the set of cloud service providers, C_i is the finite set of static workload resource pooling pattern choices of cloud service provider i, w_i is cloud service provider i's utility function. The function w_i assigns to every pair of static workload resource pooling pattern choice $(c_1, c_2) \epsilon C_1 \times C_2$ a utility $w_i(c_1, c_2) \epsilon F$.

In a static scaling with asymmetric static workload resource pooling pattern, cloud service providers do not only uncertainty about the cloud consumer's static workload resource pooling pattern choices; they also have uncertainty about the cloud consumer's utility function. Hence a conjecture hierarchy should not only specify what the cloud service provider premises about the cloud consumer's static workload resource pooling pattern choice but also what it premises about the cloud consumer's utility function. Not only this, it should also specify what the cloud service provider premises about the cloud consumer's conjecture about its own static workload resource pooling pattern choice and utility function, and so on. A possible way of modeling such conjecture hierarchies is by means of the following necessary and sufficient condition.

Necessary and sufficient condition 3.1 (*Static scaling model*). A finite static scaling model for δ with asymmetric static workload resource pooling pattern is a tuple $M = (S_i, v_i, K_i)_{i\epsilon I}$ where (1) S_i is the set of static workload types for cloud service provider i. (2) $v_i : S_i \to \theta(C_j \times S_j)$ is the conjecture assignment taking only finitely many different probability distributions on $\theta(C_j \times S_j)$ and (3) k_i is the utility assignment that assigns to every $s_i \epsilon S_i$ a utility function $k_i(s_i): C_1 \times C_2 \to F$. By $\theta(P)$ we denote the set of probability distributions on P. Therefore, in a static scaling model, each static workload type s_i has a conjecture about cloud service provider j's resource pooling pattern choice-static workload type combinations. And hence, in particular, it has a conjecture about j's resource pooling pattern choice. But, as cloud service provider j's static workload type also specifies its utility function and its conjecture about i's resource pooling pattern choice, cloud service provider i also has some conjecture about cloud service provider j's utility function, and about cloud service provider j's conjecture about its own resource pooling pattern choice, and so on. In this way one can derive a complete conjecture hierarchy for every given static workload type.

Note that each static workload type s_i can be identified with a pair $\left(k_i(s_i), v_i(s_i)\right)$ where $k_i(s_i)$ is its utility function and $v_i(s_i)$ is its conjecture hierarchy. Since we required the conjecture assignment to take only finitely many different probability distributions, the static scaling model contains only finitely many different conjecture hierarchies.

3.2 Limitations on the Static Scaling Model

Our goal will be to model the situation where the cloud service providers have uncertainty about the cloud consumer's utility function, but where this uncertainty "vanishes in the limit". In order to formalize this we need to impose additional limitations on the static scaling-model.

Recall that every static workload type s_i can be identified with a pair $\left(k_i(s_i), v_i(s_i)\right)$, where $k_i(s_i)$ is s_i's utility function and $v_i(s_i)$ is its conjecture hierarchy. Denote by K_i the set of all possible utility functions, and by V_i the set of all conjecture hierarchies in the static scaling model $M = (S_i, k_i, v_i)_{i\epsilon I}$. The first condition we impose is that $S_i = K_i \times V_i$, that is, for every possible utility function we can think of, and every conjecture hierarchy in the model, there exists a static workload type in the model with exactly this combination of utility function and conjecture hierarchy. Therefore in a sense we assume that the static workload type is rich enough.

Secondly, we assume that s_i's conjecture about j's utility from (c_1, c_2) is statistically independent from its conjecture j's utility from $(\acute{c}_1, \acute{c}_2)$ whenever $(c_1, c_2) \neq (\acute{c}_1, \acute{c}_2)$ and that this conjecture is also statistically independent from its conjecture about j's conjecture hierarchy.

Finally we assume that s_i's conjectures about j's utilities from the various outcomes in the static scaling are all induced by a unique normal distribution. More formally, s_i's conjecture about j's utility from (c_1, c_2) is given by a normal distribution with its mean at $w_j(c_1, c_2)$ – the "true" utility of cloud service provider j in the original static scaling. Therefore, all these conjectures are distributed identically around the mean. By collecting all these conditions we arrive at the following necessary and sufficient condition.

Necessary and sufficient condition 3.2 *(σ–regular static scaling model)*. Let D be the normal distribution on F with mean 0 and variance $\sigma^2 > 0$. Then a static scaling model $M = (S_i, k_i, v_i)_{i\epsilon I}$ is σ–regular if for both cloud service providers i, (1) $S_i = K_i \times V_i$, (2) for every static workload type $s_i \epsilon S_i$, its conjecture about j's utility from (c_1, c_2) is statistically independent from its conjecture about j's utility from $(\acute{c}_1, \acute{c}_2)$ whenever $(c_1, c_2) \neq (\acute{c}_1, \acute{c}_2)$ and its conjecture about j's utilities is statistically independent from its conjecture about j's conjecture hierarchy, and (3) for every static workload type $s_i \epsilon S_i$, and every resource pooling pattern choice-pair (c_1, c_2), the conjecture of s_i about j's utility from (c_1, c_2) is given by D, upto a shift of the mean to $w_j(c_1, c_2)$.

3.3 σ-Quantifiability

In this subsection we will define common conjecture in quantifiability inside a static scaling model with asymmetric static workload resource pooling pattern. In addition, if we require the static scaling-model to be σ-regular for a given normal distribution with mean 0 and variance σ^2, then we obtain the concept of σ-quantifiability. We first need some more notations. For given static workload type s_i and static workload resource pooling pattern choice c_i, let $k_i(s_i)(c_i)$ be the expected utility for static workload type s_i from choosing c_i, given its conjecture $v_i(s_i)$ about the cloud consumer's static workload resource pooling pattern choice, and given its utility function $k_i(s_i)$.

Necessary and sufficient condition 3.3 *(Quantifiable static workload resource pooling choice)*. A static workload resource pooling pattern choice c_i is quantifiable for s_i if $k_i(s_i)(c_i) \geq k_i(s_i)(\acute{c_\iota})$ for all $\acute{c_\iota} \in C_i$.

We will now define common conjecture in quantifiability. In words it says that a cloud service provider premises that its cloud consumer makes quantifiable static workload resource pooling pattern choices, and premises that its cloud consumer premises that it makes quantifiable static workload resource pooling pattern choices, and so on [25].

Formally, for every $\widehat{S_\iota} \subseteq S_i$, let

$$(C_i \times \widehat{S_\iota})^{quant} = \{(c_i, s_i) \in C_i \times \widehat{S_\iota}: c_i \text{ is quantifiable for } s_i\}.$$

Necessary and sufficient condition 3.4 *(Common conjecture in quantifiability)*. For cloud service providers i we define subsets of static workload types $S_i^1, S_i^2, \ldots$ in a recursive way as follows:

$$S_i^1 := \left\{s_i \in S_i: v_i(s_i)\left[\left(C_j \times S_j\right)^{quant}\right] = 1\right\},$$

$$S_i^2 := \left\{s_i \in S_i: v_i(s_i)\left[\left(C_j \times S_j^1\right)^{quant}\right] = 1\right\},$$

$$..S_i^l := \left\{s_i \in S_i: v_i(s_i)\left[\left(C_j \times S_j^{l-1}\right)^{quant}\right] = 1\right\},$$

Static workload type s_i expresses common conjecture in quantifiability if $s_i \in \cap_{l \in \mathbb{N}} S_i^l$. A static workload type σ–quantifiable if it expresses common conjecture in quantifiability with a σ–regular static scaling model.

Necessary and sufficient condition 3.5 *(σ–quantifiable static workload type)*. Let $M = (S_i, v_i, k_i)_{i \epsilon I}$ be a σ–regular static scaling model. Every static workload type $s_i \in S_i$ that expresses common conjecture in quantifiability is called σ–quantifiable.

Now we show that σ–quantifiable static workload types always exist.

Proposition 3.1 (σ–quantifiable static workload types always exist): Consider a finite static workload $\delta = (C_i, w_i)_{i \epsilon I}$, and some $\sigma > 0$. Then there is a σ–regular static scaling model $M = (S_i, v_i, k_i)_{i \epsilon I}$ for δ where all static workload types are σ–quantifiable. The proof can be found in Section 7.

3.4 Limit Quantifiability

In this subsection we focus on those static workload resource pooling pattern choices, which can quantifiably be made under common conjecture in quantifiability when the uncertainty about the cloud consumer's utility vanishes. This will lead to the concept of limit quantifiability. We first need an additional necessary and sufficient condition.

Necessary and sufficient condition 3.6 *(Constant static workload type and utility assignments)*. A static workload sequence of static scaling models $((S_i^n, v_i^n, k_i^n)_{i \epsilon I})_{n \epsilon \mathbb{N}}$ has constant static workload type and utility assignments if $S_i^n = S_i^m$ and $k_i^n = k_i^m$ for all n and m, and for cloud service providers i.We are now ready to say the concept of limit quantifiable static workload resource pooling pattern choice.

Necessary and sufficient condition 3.7 *(Limit quantifiable resource pooling pattern choice)*. Consider a finite static workload $\delta = (C_i, w_i)_{i \epsilon I}$ with cloud service providers. A static workload resource pooling pattern choice c_i is limit quantifiable if there is a static workload sequence $(\sigma_n)_{n \epsilon \mathbb{N}} \rightarrow 0$,and a static workload sequence $(M^n)_{n \epsilon \mathbb{N}}$ of σ_n–regular static scaling models with constant static workload type and utility assignments, such that in every M^n there is a σ_n-quantifiable static workload type s_i^n with utility function w_i, for which static workload resource pooling pattern choice c_i is optimal.

4. PRESCRIPTIVE STUDY

4.1 Static Scaling Model

Let $\delta = (C_i, w_i)_{i \epsilon I}$ be a finite, static workload with cloud service providers. In a static scaling with symmetric static workload resource pooling pattern cloud service providers do not have uncertainty about the cloud consumer's utility function. Therefore a conjecture hierarchy only needs to specify what a cloud service provider premises about the cloud consumer's static workload resource pooling pattern choice, what it premises about the cloud consumer's conjecture about its own static workload resource pooling pattern choice, and so on. Therefore the static scaling model will be simpler compared to the case of asymmetric static workload resource pooling pattern.

Necessary and sufficient condition 4.1 *(Static scaling model)*. A static scaling model for δ with symmetric static workload resource pooling pattern is a tuple $M = (\Omega_i, \rho_i)_{i \epsilon I}$ where (1) Ω_i is the finite set of static workload types for cloud service provider i, and (2) $\rho_i: \Omega_i \rightarrow \theta(C_j \times \Omega_j)$ is the conjecture assignment.

Therefore, in a static scaling model, each static workload type τ_i has a conjecture about cloud service provider j's static workload resource pooling pattern choice-static workload type combinations. And hence, in particular, it has a conjecture about j's static workload resource pooling pattern choice. But, as cloud service provider j's static workload type also specifies its conjecture about cloud service provider i's static workload resource pooling pattern choice, cloud service provider i also has some conjecture about cloud service provider j's conjecture about its own static workload resource pooling pattern choice, and so on. In this way one can derive a complete conjecture hierarchy for every given static workload type.

For given static workload type τ_i and static workload resource pooling pattern choice c_i we define $w_i(c_i, \tau_i)$ as the expected utility for static workload type τ_i from choosing c_i given its conjecture $\rho_i(\tau_i)$ about its cloud consumer's static workload resource pooling pattern choice (and *given* its "fixed" utility function w_i). Static workload type τ_i is said to *prefer* static workload resource pooling pattern choice c_i to static workload resource pooling pattern choice $\acute{c_\iota}$ when $w_i(c_i, \tau_i) > w_i(\acute{c_\iota}, \tau_i)$. We say that a static workload type τ_i *considers possible* some cloud consumer's static workload type τ_i if $\rho_i(\tau_i)(c_j, \tau_j) > 0$ for some $c_j \in C_j$.Now we introduce the key condition in actual quantifiability, which is the α–actual trembling condition. Intuitively it says that (1) a cloud service provider should deem possible all cloud consumer's static workload resource pooling pattern choices, and (2) if a cloud service provider premises static workload resource pooling pattern choice a is better than static workload resource pooling pattern choice b for the other cloud service provider, then it should deem static workload resource pooling pattern choice a much more likely than static workload resource pooling pattern choice b.

Necessary and sufficient condition 4.2 *(α-actual trembling condition)*: Let $\alpha > 0$. A static workload type τ_i satisfies the α-actual trembling condition if (1) for each τ_j that τ_i deems possible, $\rho_i(\tau_i)(c_j, \tau_j) > 0$ for all $c_j \in C_j$,and (2) for every τ_j that τ_i deems possible, whenever τ_j *prefers* c_j to $\acute{c}_j$, then $\rho_i(\tau_i)(\acute{c}_j, \tau_j) \leq \alpha \cdot \rho_i(\tau_i)(c_j, \tau_j)$.

Therefore, the first condition says that whenever τ_i deems some static workload type τ_j possible, τ_i also assumes every static workload resource pooling pattern choice is possible for τ_j. Actual quantifiability is based on the event that the static workload types should not only satisfy the α-actual trembling condition themselves, but also express common conjecture in the event that static workload types satisfy the α-actual trembling condition.

Necessary and sufficient condition 4.3 *(α-actually quantifiable static workload type)*. A static workload type τ_i is α-actually quantifiable if: τ_i satisfies the α-actual trembling condition, τ_i only deems possible cloud consumer's static workload types τ_j which satisfy the α-actual trembling condition, τ_i only deems possible cloud consumer's static workload types τ_j which only deem possible cloud service provider i's static workload types τ_i' which satisfy the α-actual trembling condition, and so on. Actually quantifiable static workload resource pooling pattern choices are those static workload resource pooling pattern choices, which can quantifiably be made by α-actually quantifiable static workload types for all α.

Necessary and sufficient condition 4.4 *(Actually quantifiable resource pooling pattern choice)*. A static workload resource pooling pattern choice c_i is α-actually quantifiable if there is a static scaling model and a α-actually quantifiable static workload type τ_i within it for which c_i is optimal. A static workload resource pooling pattern choice c_i is actually quantifiable if it is α-actually quantifiable for all $\alpha > 0$.

5. DESCRIPTIVE STUDY II

5.1 Statement of the result

For a static workload we analyzed two contexts, one with asymmetric static workload resource pooling pattern and another with symmetric static workload resource pooling pattern. In the context with asymmetric static workload resource pooling pattern, where cloud service providers have uncertainty about the cloud consumer's utility, we introduced the concept of a limit quantifiable static workload resource pooling pattern choice. In the context with symmetric static workload resource pooling pattern, where cloud service providers have no uncertainty about the cloud consumer's utility, we discussed the concept of a actually quantifiable static workload resource pooling pattern choice. In our result we connect these two concepts.

Proposition 5.1 *(Limit Quantifiability implies actual quantifiability)*: Consider a finite static workload with cloud service providers. Every limit quantifiable static workload resource pooling pattern choice for the context with asymmetric static workload resource pooling pattern is a actually quantifiable static workload resource pooling pattern choice for the context with symmetric static workload resource pooling pattern.

5.2 Illustration of the result

By means of an example we provide some intuition for our result. More precisely we show how a quantifiable static workload type in the context of asymmetric static workload resource pooling pattern can be transformed into an actually quantifiable static workload type in the context of symmetric static workload resource pooling pattern. Also we show that when σ goes to zero then ϵ goes to zero as well. Let us start with the context of asymmetric static workload resource pooling pattern. Let D be the normal distribution with mean 0 and variance σ^2. From the proof of Proposition 3.1 we know that there exists a regular static scaling model $M = (S_i, v_i, k_i)_{i \epsilon I}$ where every static workload type is quantifiable and all the static workload types have the same conjecture hierarchy. Therefore, static workload types only differ by their utility function. For each of the static workload types s_1 of cloud service provider 1 we denote by ρ_1 the conjecture about cloud service provider 2's static workload resource pooling pattern choice, and for each static workload type s_1 let ρ_1 be the conjecture about cloud service provider 1's static workload resource pooling pattern choice. As we assume that all the static workload types have the same conjecture hierarchy, ρ_1 and ρ_2 are unique.

For both cloud service providers i let O_i be the probability distribution on cloud service provider i's utility functions generated by D. Since the static scaling-model is σ-regular every static workload type s_j has the conjecture O_i about i's utility function. Let $K_i(c_i, \rho_i)$ be the set of utility functions for cloud service provider i such that the static workload resource pooling pattern choice c_i is optimal under the conjecture ρ_i about the cloud consumer's static workload resource pooling pattern choice. Since every static workload type s_i expresses common conjecture in quantifiability, the probability it assigns to a cloud consumer's static workload resource pooling pattern choice c_j is exactly the probability it assigns to the event that j's utility function is in $K_j(c_j, \rho_j)$ which is $O_j\left(K_j(c_j, \rho_j)\right)$.

Since D has full support, it follows that all these probabilities are positive. Now we turn to the context of symmetric static workload resource pooling pattern. We construct a static scaling model with a single static workload type τ_1 for cloud service provider 1 and a single static workload type τ_2 for cloud service provider 2. Let the conjecture of τ_1 about the cloud service provider 2's static workload resource pooling pattern choice be given by the ρ_1 constructed above, and similarly for the conjecture of τ_2. Therefore, the conjecture about the cloud consumer's static workload resource pooling pattern choice has not changed by moving from the context with asymmetric static workload resource pooling pattern to the context with symmetric static workload resource pooling pattern.

6. CONCLUDING REMARKS

We premise that actual quantifiability is a very natural concept in static scaling, but it has not yet received the attention it deserves. In this paper we have established a new approach for actual quantifiability from the viewpoint of static scaling with asymmetric static workload resource pooling pattern. In static scaling with asymmetric static workload resource pooling pattern we define a static workload resource pooling pattern choice as limit quantifiable if it can quantifiably be made under common conjecture of quantifiability when the uncertainty vanishes gradually in some regular way. We show the existence of such static workload resource pooling pattern choices. We then prove that each limit quantifiable static workload resource pooling pattern choice in the static scaling with asymmetric static workload resource pooling pattern is actually quantifiable for the context with symmetric static workload resource pooling pattern.

7. PROOFS

7.1 Existence of Quantifiable Static Workload Types

We prove Proposition 3.1, which guarantees the existence of σ-quantifiable static workload types. Consider a finite static workload $M = (C_i, w_i)_{i\epsilon I}$ and, some $\sigma > 0$. Let D be the normal distribution with mean 0 and variance σ^2. In fact we will construct a σ-regular static scaling model where all static workload types of cloud service provider 1 have the same conjecture ρ_2 about cloud service provider 2's static workload resource pooling pattern choice and all static workload types of cloud service provider 2 have the same conjecture ρ_1 about cloud service provider 1's static workload resource pooling pattern choice. We construct ρ_1 and ρ_2 by means of the fixed key of some correspondence.

For every conjecture $\rho_j \in \theta(C_j)$ and every utility function w_i, we define

$$C_i(\rho_j, w_i) := \{C_i \in C_i : w_i(c_j, \rho_j) \geq w_i(\acute{c_\iota}, \rho_j) \text{ for all } \acute{c_\iota}\}.$$

We also define O_i as the probability distribution on the set of utility functions of cloud service provider i induced by D. For every $\rho_j \in \theta(C_j)$ we define

$$G_i(\rho_j) := \{\rho_i \in \theta(C_j) : \rho_i = \int_{w_i \in K_i} \phi_i(x_i)\ dO_i,$$

where $\phi_i(x_i) \in \left(C_i(\rho_j, x_i)\right)$ for every $x_i \in K_i\}$.

Here K_i denotes the set of all possible utility functions for cloud service provider i. Therefore every $\rho_i \in G_i(\rho_j)$ is obtained by taking for every utility function x_i a randomization over optimal static workload resource pooling pattern choices against ρ_j and then taking the expected randomization with respect to O_i. Now we define a correspondence G from $\theta(C_1) \times \theta(C_2)$ to $\theta(C_1) \times \theta(C_2)$ by

$$G(\rho_1, \rho_2) := G_1(\rho_2) \times G_2(\rho_1).$$

Now we use fixed key position to prove that G has a fixed key. Clearly G is upper hemi-continuous and compact valued. We show that G is convex valued. For this it is sufficient to show that $G_1$and G_2 are convex valued. For a given ρ_2, take ρ_1', ρ_1'' in (ρ_2). We show that $\psi\rho_1' + (1-\psi)\rho_1''$ is also in $G_1(\rho_2)$. By definition

$$\rho_1' = \int_{x_1} \phi_1'(x_1)\, dO_1 \text{ and } \rho_1'' = \int_{x_1} \phi_1''(x_1)\ dO_1$$

where $\phi_1'(x_1)$, $\phi_1''(x_1) \in \theta\left(C_1(\rho_2, x_1)\right)$ for every x_1. Therefore we have

$$\psi\rho_1' + (1-\psi)\rho_1'' = \int_{x_1} (\psi\phi_1'(x_1) + (1-\psi)\phi_1''(x_1))\, dO_1$$

where $\psi\phi_1'(x_1) + (1-\psi)\phi_1''(x_1) \in \theta\left(C_1(\rho_2, x_1)\right)$ for every x_1. Hence by definition $\psi\rho_1' + (1-\psi)\rho_1'' \in G_1(\rho_2)$. This implies that G_1 is convex valued. The same applies to G_2 and hence we can conclude that G is convex valued. Now using fixed key position G has a fixed key (ρ_1^*, ρ_1^*).

Since $\rho_1^* \in G_1(\rho_2^*)$ it follows that

$$\rho_1^* = \int_{x_1} \phi_1^*(x_1)\ dO_1$$

where $\phi_1^*(x_1) \in \theta\left(C_1(\rho_2^*, x_1)\right)$ for every x_1. Similarly

$$\rho_2^* = \int_{x_2} \phi_2^*(x_2)\ dO_2$$

where $\phi_2^*(x_2) \in \theta\left(C_2(\rho_1^*, x_2)\right)$ for every x_2.

We will now construct a static scaling model $M = (S_i, v_i, k_i)_{i\epsilon I}$. For both cloud service providers i, define

$$S_i = \{{s_i}^{x_i} : x_i \in K_i\}.$$

Let the utility assignment k_i be given by

$$k_i({s_i}^{x_i}) = x_i$$

for every ${s_i}^{x_i} \in S_i$. In order to define the conjecture assignment v_i we first define for every static workload type ${s_i}^{x_i}$ a density function $v_i^{\sim}({s_i}^{x_i})$ on $C_j \times S_j$ as follows: $v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j}) := \phi_j^*(x_j)(c_j)$,where $\phi_j^*(x_j)(c_j)$ is the probability that probability distribution $\phi_j^*(x_j)$ assigns to c_j. For every static workload type ${s_i}^{x_i}$ let $v_i({s_i}^{x_i}) \in \theta(C_j \times S_j)$ be the probability distribution induced by density function $v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j})$ and the probability distribution Q_j on K_j. That is, for every set of static workload types $H \subseteq S_j$given by

$$H := \{{s_j}^{x_j} : x_j \in G\}$$

We have that

$$v_i({s_i}^{x_i})(\{c_j\} \times H) := \int_{x_j \in G} v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j})\ dO_j$$

It follows that the conjecture of static workload type ${s_i}^{x_i}$ about cloud service provider j's resource pooling pattern choice is given by ρ_j^*. Namely, the probability that static workload type ${s_i}^{x_i}$ assigns to static workload resource pooling pattern choice c_j is equal to

$$v_i({s_i}^{x_i})(\{c_j\} \times K_j) = \int_{x_j \in K_j} v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j})\ dO_j$$

$$= \int_{x_j \in K_j} \phi_j^*(x_j)(c_j)\ dO_j$$

$$= \rho_j^*(c_j).$$

Therefore all static workload types of cloud service provider i have the same conjecture ρ_j^* about cloud service provider j's static workload resource pooling pattern choice. This completes the construction of the static scaling model. It follows directly from the construction that the static scaling model is σ-regular.We now show that every static workload type in this model expresses common conjecture in quantifiability. For this it is sufficient to show that every static workload type ${s_i}^{x_i}$ premises in the cloud consumer's quantifiability. Therefore, we must show for the cloud service providers i and every ${s_i}^{x_i} \epsilon S_i$ that $v_i^{\sim}({s_i}^{x_i})[(C_j \times S_j)^{quant}] = 1$. In order to prove, we show that $v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j}) > 0$ only if c_j is quantifiable for ${s_j}^{x_j}$.

Suppose that $v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j}) > 0$. Since $v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j}) \coloneqq \phi_j^*(x_j)(c_j)$, it follows that $\phi_j^*(x_j)(c_j) > 0$. As by definition $\phi_j^*(x_j) \epsilon\, \theta\left(C_j(\rho_i^*, x_j)\right)$ it follows that $c_j \epsilon\, C_j(\rho_i^*, x_j)$. Remember that the conjecture of static workload type ${s_j}^{x_j}$ about cloud service provider i's static workload resource pooling pattern choice is exactly ρ_i^*. Since $c_j \epsilon\, C_j(\rho_i^*, x_j)$ it follows that c_j is quantifiable for static workload type ${s_j}^{x_j}$. Therefore we have shown that $v_i^{\sim}({s_i}^{x_i})(c_j, {s_j}^{x_j}) > 0$ only if c_j is quantifiable for ${s_j}^{x_j}$. This implies that static workload type ${s_i}^{x_i}$ premises in the cloud consumer's quantifiability. Since this holds for every static workload type in the model it follows that every static workload type in the static scaling model expresses common conjecture in quantifiability. Therefore every static workload type in the model is σ-quantifiable because the model is σ-regular. This completes the proof.

7.2 Corollaries

In this subsection we state some technical corollaries, which we need for the proof of the result.

Corollary 7.1. *If P, Q and Rare data valued, independent random variables then* $\Pr(P \geq \max\{Q, R\}) \geq \Pr(P \geq Q) \cdot \Pr(P \geq R)$.

Proof. Let g_Q and g_R be the probability density functions of the random variables Q and R.

Now,

$$\Pr(P \geq \max\{Q, R\})$$

$$= \int_q \int_r \Pr(P \geq \max\{q, r\})\ dg_Q(q)\ dg_R(r)$$

$$\geq \int_q \int_r \Pr(P \geq \max\{q, r\}) \cdot \Pr(P \geq \min\{q, r\})\ dg_Q(q)\ dg_R(r)$$

$$= \int_q \int_r \Pr(P \geq q)\ \cdot \Pr(P \geq r)\, dg_Q(q)\, dg_R(r)$$

$$= \int_q \Pr(P \geq q)\ dg_Q(q) \cdot \int_r \Pr(P \geq r)\ dg_R(r)$$

$$= \Pr(P \geq Q) \cdot \Pr(P \geq R).$$

Note that the first and third equality follow from the fact that Q and R are independent, and the inequality holds because $\Pr(P \geq \min\{q, r\}) \leq 1$.

Corollary 7.2. *Let P be a random variable with $H(P)$=γ. Then for any number $t > 0$,*

$$(|P - \gamma| \geq t) \leq \frac{\mathrm{Var}(P)}{t^2}$$

Corollary 7.3. *For every $n \epsilon \mathbb{N}$, let $P_n^1, P_n^2, \ldots P_n^m$be independent random variables with $H(P_n^i)$=γ^i for all n and i, $\gamma^1 > \gamma^2 > \cdots > \gamma^m$, and* $\lim_{n\to\infty} \mathrm{Var}(P_n^i) = 0$ *for all i. Then,*

$$\lim_{n\to\infty} \Pr(P_n^1 \geq P_n^2 \geq \cdots\ \geq P_n^m) = 1.$$

Proof. For a given n,

$$\Pr(P_n^1 \geq P_n^2 \geq \cdots\ \geq P_n^m) \geq 1 - \Pr(P_n^i \geq P_n^j \text{for some } i < j).$$

For fixed $i < j$ we have,

$$\Pr(P_n^i < P_n^j) = \Pr(P_n^j - P_n^i > 0)$$

$$= \Pr\left((P_n^j - P_n^i) - (\gamma^j - \gamma^i) > \gamma^i - \gamma^j\right)$$

$$\leq \Pr\left(\left|(P_n^j - P_n^i) - (\gamma^j - \gamma^i)\right| > \gamma^i - \gamma^j\right)$$

$$\leq \frac{\mathrm{Var}(P_n^j - P_n^i)}{(\gamma^i - \gamma^j)^2}$$

$$= \frac{\mathrm{Var}(P_n^j) + \mathrm{Var}(P_n^i)}{(\gamma^i - \gamma^j)^2}$$

The last equality follows from the fact that P_n^j and P_n^i are independent. Now, note that $\lim\limits_{n\to\infty} \mathrm{Var}(P_n^i) = 0$ and $\lim\limits_{n\to\infty} \mathrm{Var}(P_n^j) = 0$, which implies $\lim\limits_{n\to\infty} \mathrm{Var}(P_n^i < P_n^j) = 0$. Then, from above it follows that

$$\lim_{n\to\infty} \Pr(P_n^1 \geq P_n^2 \geq \cdots\ \geq P_n^m) = 1.$$

Consider a static workload sequence $(D_n)_{n\epsilon\mathbb{N}}$ of normal distributions with mean 0 and variance σ_n^2 such that $\sigma_n \to 0$ as $n \to \infty$. The density function $g_n$of D_n is given by

$$g_n(p) = \frac{2}{\sigma_n\sqrt{2\pi}}\ \mathrm{e}^{-(p^2/2\sigma_n^2)} \text{ for all } p\ .$$

We show that for large n the right tail of D_n becomes arbitrarily steep everywhere.

Corollary 7.4. *Consider a static workload sequence $(D_n)_{n\epsilon\mathbb{N}}$ of normal distributions with mean 0 and variance σ_n^2 , such that $\sigma_n \to 0$ as $n \to \infty$. Let g_n be the density functions of these distributions. Then for all $c > 0$ and $\alpha > 0$ there is $N \epsilon \mathbb{N}$ such that $g_n(p + c)/g_n(p) \leq \alpha$ for all $n \geq N$ and all $p > 0$.*

Proof. Take $c > 0$ and $\alpha > 0$. Then

$$\frac{g_n(p + c)}{g_n(p)} = \frac{\mathrm{e}^{-((p+c)^2/2\sigma_n^2)}}{\mathrm{e}^{-(p^2/2\sigma_n^2)}} = \mathrm{e}^{-(1/2\sigma_n^2)((p+c)^2 - p^2)}$$

$$= \mathrm{e}^{-(1/2\sigma_n^2)(2cp + c^2)} \leq \mathrm{e}^{-(c^2/2\sigma_n^2)}$$

Now as $c > 0$ is fixed and $\sigma_n \to 0$ as $n \to \infty$, we can find N large enough such that $\mathrm{e}^{-(c^2/2\sigma_n^2)} \leq \alpha$ for $n \geq N$.

Corollary 7.5. *Consider a static workload sequence $(D_n)_{n\epsilon\mathbb{N}}$ of normally distributed random variables such that $H(P_n) = 0$ for all n, and $var(P_n) \to 0$ as $n \to \infty$. Let g_n be the density functions of these random variables. Then, for every $0 < p < q$ it holds that*

$$\lim_{n\to\infty} \frac{\Pr(P_n \geq q)}{\Pr(P_n \geq p)} = 0\ .$$

Proof. Fix $0 < p < q$ and fix a $\alpha > 0$. Then, by corollary 7.4 there is an N such that $g_n(r + (q - p))/g_n(r) \leq \alpha$ for all $n \geq N$ and all $r > 0$. Take some $n \geq N$. Then,

$$\Pr(P_n \geq q) = \int_q^\infty g_n(r)\,dr = \int_p^\infty g_n(r + (q - p))\,dr$$

$$\leq \alpha \cdot \int_p^\infty g_n(r)\,dr = \alpha \cdot \Pr(P_n \geq p).$$

This implies that

$$\lim_{n\to\infty} \frac{\Pr(P_n \geq q)}{\Pr(P_n \geq p)} = 0\,.$$

7.3 Proof of the result

We finally prove our main proposition, which is Proposition 5.1. We proceed by three steps.

In step 1, we show how a σ-regular static scaling model M with asymmetric static workload resource pooling pattern can be transformed into a static scaling model M' with symmetric static workload resource pooling pattern. More precisely, we transform every static workload type s_i in M into a static workload type $\tau_i(s_i)$ in M' which has the same conjecture about the cloud consumer's static workload resource pooling pattern choice as s_i. In step 2, we take a static workload resource pooling pattern choice c_i^* that is limit quantifiable. Therefore we can find a static workload sequence $(D_n)_{n\epsilon\mathbb{N}}$ of normal distributions with mean 0 and variance σ_n^2, with $\sigma_n^2 \to 0$ as $n \to \infty$, and a static workload sequence $(M^n)_{n\epsilon\mathbb{N}}$ of σ_n-regular static scaling models with constant static workload type and utility assignments, such that in every M^n there is a σ_n-quantifiable static workload type s_i^n with utility function w_i for which resource pooling static workload pattern choice c_i^* is optimal. We show that the static workload type s_i^n is transformed into a static workload type $\tau_i(s_i^n)$ which is α_n-actually quantifiable for some α_n. Since, for all n, c_i^* is quantifiable for t_i^n and $\tau_i(s_i^n)$ has the same conjecture about the cloud consumer's static workload resource pooling pattern choice and the same utility function as s_i^n, it follows that c_i^* is quantifiable for $\tau_i(s_i^n)$ for all n. As $\tau_i(s_i^n)$ is α_n-actually quantifiable for every n, it follows that c_i^* is α_n-actually quantifiable for all n. In step 3, we prove that $\lim_{n\to\infty} \alpha_n = 0$. Hence, c_i^* is α-actually quantifiable for every $\alpha > 0$ and therefore actually quantifiable.

Step 1. Take some $\sigma > 0$. Let $M = (S_i, v_i, k_i)_{i\epsilon I}$ be a σ-regular static scalingmodel for δ with asymmetric static workload resource pooling pattern. Now we transform this static scaling model M into a static scaling model $M' = (\Omega_i, \rho_i)_{i\epsilon I}$ with symmetric static workload resource pooling pattern. Using the fact that M is σ-regular we can write

$$v_i(s_i) \in \theta\big(C_j \times k_j \times V_j\big).$$

Now take $\Omega_i = V_i$ and $\Omega_j = V_j$. Clearly, Ω_i and Ω_j are finite sets as V_i and V_j are finite. For every $s_i \in S_i$ define the static workload type $\tau_i(s_i)$ by

$$\rho_i\big(\tau_i(s_i)\big) \coloneqq \max g_{C_j \times V_j} v_i(s_i).$$

Therefore,

$$\rho_i\big(\tau_i(s_i)\big)\big(C_j \times V_j\big) = v_i(s_i)\,\big(K_j \times \{(C_j \times V_j)\}\big)$$

for all $\big(C_j \times V_j\big)$. Hence,

$$\rho_i\big(\tau_i(s_i)\big) \in \theta\big(C_j \times V_j\big) = \theta\big(C_j \times \Omega_j\big)$$

By construction $\tau_i(s_i)$ has the same conjecture about j's static workload resource pooling pattern choice as s_i. This completes the construction of the static scaling $M' = (\Omega_i, \rho_i)_{i\epsilon I}$.

Step 2. Take a static workload resource pooling pattern choice c_i^* that is limit quantifiable. Hence, there exists a static workload sequence $(D_n)_{n\epsilon\mathbb{N}}$ of normal distributions with mean 0 and variance σ_n^2, with $\sigma_n^2 \to 0$ as $n \to \infty$, and a static workload sequence $(M^n)_{n\epsilon\mathbb{N}}$ of σ_n-regular static scaling models with constant static workload type and utility assignments, such that in every M^n there is a σ_n-quantifiable static workload type s_i^n with utility function w_i for which static workload resource pooling pattern choice c_i^* is optimal. Let the constant static workload type in the static workload sequence $(M^n)_{n\epsilon\mathbb{N}}$ of static scaling models be S_i and S_j, and the constant utility assignments be $k_i$and k_j.Fix an n. Then, within the static scaling model $M^n = (S_i, v_i^n, k_i)_{i\epsilon I}$ there is an σ_n-quantifiable static workload type $s_i^n \in S_i$ with utility function w_i for which c_i^* is optimal. Since static workload type s_i^n only deems possible j's static workload types which are σ_n-quantifiable, and only deems possible j's static workload types which only deem possible i's static workload types which are σ_n-quantifiable and so on. We may assume without loss of generality that all the static workload types in M^n are σ_n-quantifiable. Let $M'^n = (\Omega_i^n, \rho_i^n)_{i\epsilon I}$ be the corresponding static scaling model with symmetric static workload resource pooling pattern, as constructed in step 1. For every $\tau_i \in \Omega_i^n$, we define a number $\alpha_n(\tau_i)$ as follows: Let $\mathrm{Poss}(\tau_i)$ be the set of static workload types in Ω_j that Ω_i deems possible. For a given static workload type $\tau_j \in \mathrm{Poss}(\tau_i)$, suppose that τ_j prefers static workload resource pooling pattern choice c_j^1 to c_j^2, c_j^2 to c_j^3, and so on. Therefore, we obtain an ordering $(c_j^1, c_j^2, c_j^3, \ldots c_j^m)$ of j's static workload resource pooling pattern choices.

Then define

$$\alpha_n\big(\tau_i, \tau_j\big) = \max_{t \in \{2,3,\ldots,m\}} \frac{\rho_i^n(\tau_i)(c_j^t, \tau_j)}{\rho_i^n(\tau_i)(c_j^{t-1}, \tau_j)}$$

Next we define

$$\alpha_{i,n} = \max_{\tau_i \in \Omega_i^n, \tau_j \in \mathrm{Poss}(\tau_i)} \alpha_n\big(\tau_i, \tau_j\big)$$

Finally let

$$\alpha_n = \max\{\alpha_{i,n}, \alpha_{j,n}\}.$$

Note that by construction every static workload type in M'^n satisfies the α_n-actual trembling condition; hence every static workload type in M'^n is α_n-actually quantifiable. In particular $\tau_i(s_i^n)$ is α_n-actually quantifiable [19] [26].

Step 3. Now we show that $\lim_{n\to\infty} \alpha_n = 0$. It is sufficient to show that

$$\lim_{n\to\infty} \frac{\rho_i^n(\tau_i)(c_j^t, \tau_j)}{\rho_i^n(\tau_i)(c_j^{t-1}, \tau_j)} = 0 \qquad (1)$$

for every $\tau_i \in \Omega_i^n$ and every $\tau_j \in \mathrm{Poss}(\tau_i)$ and every t. As before, cloud service provider j's static workload resource pooling pattern choices are ordered $c_j^1, \ldots c_j^m$ such that τ_j prefers static workload resource pooling pattern choice c_j^1 to c_j^2, c_j^2 to c_j^3, and so on. We assume, without loss of generality, that all resource pooling pattern preferences are strict. Fix some $\tau_i \in \Omega_i^n$ and $\tau_j \in \mathrm{Poss}(\tau_i)$. Suppose that $\tau_i = \tau_i(s_i)$ for some $s_i \in S_i$ and that $\tau_j = \tau_j(s_j)$ for some $s_j \in S_j$. Let $\phi_j \in \theta(C_i)$ be τ_i's conjecture about i's static workload resource pooling pattern choice. As before, let K_j be the set of utility functions for cloud service provider j. For every $t \in \{1, \dots, m\}$, let $P^t\colon K_j \to F$ be given by

$$P^t(k_j) \coloneqq k_j(c_j^t, \phi_j) = \sum_{c_i \in C_i} \phi_j(c_i) \cdot k_j(c_j^t, c_i)$$

for every $k_j \in K_j$. Therefore, $P^t(k_j)$ denotes the expected utility for cloud service provider j induced by static workload resource pooling pattern choice c_j^t, under the conjecture ϕ_j and the utility function k_j. Note that P^t is a random variable, as cloud service provider i holds a probability distribution on K_j, induced by D. The probability distribution of P^t depends on n, and is denoted by $\omega_n^t(P^t)$. Note that P^t has a normal distribution with mean

$$H(P^t) = w_j(c_j^t, \phi_j),$$

and variance

$$Var^n(P^t) = \textstyle\sum_{c_i \in C_i} (\phi_j(c_i))^2 \cdot \sigma_n^2 \qquad (2)$$

In particular, it follows that $\lim_{n} \to_\infty Var^n(P^t) = 0$, as $\lim_n \to_\infty \sigma_n^2$. Since, by assumption, τ_j strictly prefers c_j^1 to c_j^2, strictly prefers c_j^2 to c_j^3, and so on, we have that $H(P^1) > H(P^2) > \cdots > H(P^m)$.Let ω^n be the probability distribution of the random set of data value $(P^1, \dots, P^m)$ [6] [14] [18]. Recall that all static workload types in M^n are σ_n-quantifiable, which implies that all static workload types in M^n express common conjecture in quantifiability. As such, static workload type $s_i \in S_i$ (which generates τ_i) expresses common conjecture in quantifiability. In particular, s_i only assigns positive probability to those static workload resource pooling pattern choice-static workload type combinations (c_j, s_j) where c_j is optimal for t_j. Now, as $\tau_i = \tau_i(s_i)$ and $\tau_j = \tau_j(s_j)$, we have that $\rho_i^n(\tau_i)(c_j^t, \tau_j)$ is the probability that c_j^t is optimal for s_j, and that is $\omega^n(P^t \geq P^l \text{for all } l)$. Then,

$$\frac{\rho_i^n(\tau_i)(c_j^t, \tau_j)}{\rho_i^n(\tau_i)(c_j^{t-1}, \tau_j)} = \frac{\omega^n(P^t \geq P^l \text{ for all } l)}{\omega^n(P^{t-1} \geq P^l \text{ for all } l)} \qquad (3)$$

Hence, in order to prove (1), we must show that

$$\lim_{n \to \infty} \frac{\omega^n(P^t \geq P^l \text{ for all } l)}{\omega^n(P^{t-1} \geq P^l \text{ for all } l)} = 0$$

for all $t \in \{2, \dots, m\}$,. We distinguish two cases.

Case 1. First we consider the case where $t = 2$. Then we have,

$$\frac{\omega^n(P^t \geq P^l \text{ for all } l)}{\omega^n(P^{t-1} \geq P^l \text{ for all } l)} \leq \frac{\omega^n(P^2 \geq P^1)}{\omega^n(P^1 \geq P^2 \geq P^3 \geq \cdots \geq P^m)}$$

Recall that $H(P^1) > H(P^2) > \cdots > H(P^m)$. But then, by Corollary 7.3, $\omega^n(P^2 \geq P^1) \to 0$ and $\omega^n(P^1 \geq P^2 \geq P^3 \geq \cdots \geq P^m) \to 1$, and hence

$$\frac{\omega^n(P^2 \geq P^1)}{\omega^n(P^1 \geq P^2 \geq P^3 \geq \cdots \geq P^m)} \to 0\,,$$

which implies that

$$\frac{\omega^n(P^t \geq P^l \text{ for all } l)}{\omega^n(P^{t-1} \geq P^l \text{ for all } l)} \to 0\,,$$

as $n \to \infty$.

Case 2. Now we consider the case where $t > 2$.

Let P^{max} be the random variable given by $P^{max} \coloneqq \max_{j \neq t, t-1} P_j$. We have

$$\frac{\omega^n(P^t \geq P^l \text{ for all } l)}{\omega^n(P^{t-1} \geq P^l \text{ for all } l)}$$

$$= \frac{\omega^n((P^t \geq P^{t-1}) \text{ and} (P^t \geq P^{max}))}{\omega^n((P^{t-1} \geq P^t) \text{ and} (P^{t-1} \geq P^{max}))}$$

$$\leq \frac{\omega^n(P^t \geq P^{max})}{\omega^n((P^{t-1} \geq P^t) \text{ and} (P^{t-1} \geq P^{max}))}$$

$$\leq \text{(by Corollary 7.1)} \; \frac{\omega^n(P^t \geq P^{max})}{\omega^n((P^{t-1} \geq P^t) \cdot \omega^n(P^{t-1} \geq P^{max}))}$$

$$= \frac{\omega^n(P^t \geq P^{max})}{\omega^n(P^{t-1} \geq P^{max})} \cdot \frac{1}{\omega^n(P^{t-1} \geq P^t)}$$

$$= \frac{\omega^n(P^t \geq P^{max})}{\omega^n(P^{t-1} \geq P^{max} - (H(P^{t-1}) - H(P^t)))} \cdot \frac{1}{\omega^n(P^{t-1} \geq P^t)}$$

where the last equality follows from the observation that $P^{t-1} - H(P^{t-1})$ and $P^t - H(P^t)$ have the same distribution.

Now, from Corollary 7.3 it follows that $\omega^n(P^{t-1} \geq P^t) \to 1$ as $n \to \infty$.

We show that

$$\frac{\omega^n(P^t \geq P^{max})}{\omega^n\left(P^{t-1} \geq P^{max} - \left(H(P^{t-1}) - H(P^t)\right)\right)} \to 0$$

as $n \to \infty$.

Let us define $c\colon H(P^{t-1}) - H(P^t)$. Therefore, we have to show that

$$\frac{\omega^n(P^t \geq P^{max})}{\omega^n(P^{t-1} \geq P^{max} - c)} \to 0 \qquad (4)$$

as $n \to \infty$. Note that $\omega^n(P^t \geq P^{max}) \leq \omega^n(P^t \geq P^1)$. We first show that there exists $N \in \mathbb{N}$ such that for all $n \in N$,

$$\omega^n(P^t \geq P^{max} - c) \geq \omega^n(P^t \geq P^1 - c/2) \qquad (5)$$

Now,

$$\omega^n(P^t \geq P^{max} - c)$$

$$= \omega^n(P^t \geq P^{max} - c \mid P^{max} = P^1) \cdot \omega^n(P^{max} = P^1)$$

$$+ \omega^n(P^t \geq P^{max} - c \mid P^{max} \neq P^1) \cdot \omega^n(P^{max} \neq P^1)$$

$$\geq \omega^n(P^t \geq P^{max} - c | P^{max} = P^1) \cdot \omega^n(P^{max} = P^1)$$

$$= \omega^n(P^t \geq P^1 - c) \cdot \omega^n(P^{max} = P^1)$$

Therefore, to show (5) it is sufficient to show that there exists $N \in \mathbb{N}$ such that for all $n \geq N$,

$$\omega^n(P^t \geq P^1 - c) \cdot \omega^n(P^{max} = P^1) \geq \omega^n(P^t \geq P^1 - c/2) \qquad (6)$$

Using Corollary 7.3, $\omega^n(P^{max} = P^1) \to 1$ as $n \to \infty$. We have,

$$\frac{\omega^n(P^t \geq P^1 - c/2)}{\omega^n(P^t \geq P^1 - c)}$$

$$= \frac{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq -c/2 - (H(P^t) - H(P^1)))}{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq -c - (H(P^t) - H(P^1)))}$$

Note that $\omega^n((P^t - P^1) - (H(P^t) - H(P^1)))$ has a normal distribution with mean 0 and where the variance of $\omega^n(P^t - P^1)$ tends to 0 as $n \to \infty$. Moreover, $-c - (H(P^t) - H(P^1)) > 0$ as $H(P^t) - H(P^1) < H(P^t) - H(P^{t-1}) = -c$. Hence, using Corollary 7.5,

$$\frac{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq -c/2 - (H(P^t) - H(P^1)))}{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq -c - (H(P^t) - H(P^1)))}$$
$$\to 0$$

as $n \to \infty$. Then, we have,

$$\frac{\omega^n(P^t \geq P^1 - c/2)}{\omega^n(P^t \geq P^1 - c)} \to 0$$

Therefore, there exists $N \in \mathbb{N}$ such that for all $n \geq N$,

$$\omega^n(P^{max} = P^1) \geq \frac{\omega^n(P^t \geq P^1 - c/2)}{\omega^n(P^t \geq P^1 - c)}$$

This proves (6), which as we have shown, implies (5). Now, by (5) we have

$$\frac{\omega^n(P^t \geq P^{max})}{\omega^n(P^t \geq P^1 - c)}$$
$$\leq \frac{\omega^n(P^t \geq P^1)}{\omega^n(P^t \geq P^1 - c/2)}$$
$$= \frac{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq -(H(P^t) - H(P^1)))}{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq -c/2 - (H(P^t) - H(P^1)))}$$
$$= \frac{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq (H(P^t) - H(P^1)))}{\omega^n((P^t \geq P^1) - (H(P^t) - H(P^1)) \geq (H(P^t) - H(P^1)))}$$
$$\to 0$$

as n goes to infinity. Here the convergence follows from Corollary 7.5 as $(H(P^1) - H(P^t)) \geq -c/2$. Therefore, we have shown (4), which completes case 2. Hence, we have shown that (1) holds for all t. Therefore, $\lim_{n\to\infty} \alpha = 0$ and hence the proof is complete.

8. REFERENCES

[1] Kiran M., Saikat Mukherjee, Ravi Prakash G., Characterization of Randomized Shuffle and Sort Quantifiability in MapReduce Model, International Journal of Computer Applications, 51-58, Volume 79, No. 5, October 2013.

[2] Amresh Kumar, Kiran M., Saikat Mukherjee, Ravi Prakash G., Verification and Validation of MapReduce Program model for Parallel K-Means algorithm on Hadoop Cluster, International Journal of Computer Applications, 48-55, Volume 72, No. 8, June 2013.

[3] Barroso, L.A., Ho¨lzle, U.: The datacenter as a computer: an introduction to the design of warehouse-scale machines. Synth. Lect. Comput. Architect. 4, 1–45 (2009).

[4] Kiran M., Amresh Kumar, Saikat Mukherjee, Ravi Prakash G., Verification and Validation of MapReduce Program Model for Parallel Support Vector Machine Algorithm on Hadoop Cluster, International Journal of Computer Science Issues, 317-325, Vol. 10, Issue 3, No. 1, May 2013.

[5] Ravi Prakash G, Kiran M. Saikat Mukherjee, On Randomized Preference Limitation Protocol for Quantifiable Shuffle and Sort Behavioral Implications in MapReduce Programming Model, Parallel & Cloud Computing, Vol. 3, Issue 1, 1-14, January 2014.

[6] Fehling, C., Leymann, F., Mietzner, R., Schupeck, W.: A collection of patterns for cloud types, cloud service models, and cloud-based application architectures. Technical report, University of Stuttgart (2011)

[7] Ravi Prakash G, Kiran M, On The Least Economical MapReduce Sets for Summarization Expressions, International Journal of Computer Applications, 13-20, Volume 94, No.7, May 2014.

[8] Ravi (Ravinder) Prakash G, Kiran M., On Randomized Minimal MapReduce Sets for Filtering Expressions, International Journal of Computer Applications, Volume 98, No. 3, Pages 1-8, July 2014.

[9] Fehling, C., Leymann, F., Retter, R., Schumm, D., Schupeck, W.: An architectural pattern language of cloud-based applications. In: Proceedings of the 18th Conference on Pattern Languages of Programs (PLoP), Portland, (2011).

[10] Fehling, C., Leymann, F., Rutschlin, J., Schumm, D.: Pattern-based development and management of cloud applications. Future Internet 4, 110–141 (2012). (doi:10.3390/fi4010110)

[11] Ravi (Ravinder) Prakash G, Kiran M., How Minimal are MapReduce Arrangements for Binning Expressions. International Journal of Computer Applications Volume 99 (11): 7-14, August 2014.

[12] Ravi (Ravinder) Prakash G, Kiran M., Shuffling Expressions with MapReduce Arrangements and the Role of Binary Path Symmetry. International Journal of Computer Applications 102(16): 19-24, September 2014.

[13] Dimitri P. Bertsekas and John N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, Hardcover Edition (appeared in 2015), ISBN: 1-886529-15-9 Publication: 2015, 735 pages.

[14] Ravi (Ravinder) Prakash G, Kiran M; How Replicated Join Expressions Equal Map Phase or Reduce Phase in a MapReduce Structure? International Journal of Computer Applications, Volume 107 (12): 43-50, December 2014.

[15] Fehling, C., Ewald, T., Leymann, F., Pauly, M., Ru¨tschlin, J., Schumm, D.: Capturing cloud computing knowledge and experience in patterns. In: Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD), Honolulu, (2012).

[16] Bauer, E., Adams, R.: Reliability and Availability of Cloud Computing. Wiley-IEEE Press, Hoboken (2012).

[17] Ravi (Ravinder) Prakash G, Kiran M., On Composite Join Expressions of Map-side with many Reduce Phase. International Journal of Computer Applications Volume 110(9): 37-44, January 2015.

[18] Dimitri P. Bertsekas, Convex Optimization Algorithms, Athena Scientific, Hardcover Edition ISBN: 1-886529-28-0, 978-1-886529-28-1, Publication: February, 2015, 576 pages.

[19] Ravi (Ravinder) Prakash G, Kiran M; How Reduce Side Join Part File Expressions Equal MapReduce Structure into Task Consequences, Performance? International Journal of Computer Applications, Volume 105(2):8-15, November 2014

[20] Ravi (Ravinder) Prakash G, Kiran M. "On the MapReduce Arrangements of Cartesian product Specific Expressions". International Journal of Computer Applications 112(9):34-41, February 2015.

[21] Ravi (Ravinder) Prakash G, Kiran M., On Job Chaining MapReduce Meta Expressions of Mapping and Reducing Entropy Densities. International Journal of Computer Applications 113(15): 20-27, March 2015.

[22] Ravi (Ravinder) Prakash G, Kiran M. "On Chain Folding Problems of Chain Mapper and Chain Reducer Meta Expressions". International Journal of Computer Applications 116(16): 35-42, April 2015.

[23] Ravi (Ravinder) Prakash G, Kiran M."On Job Merging MapReduce Meta Expressions for Multiple Decomposition Mapping and Reducing". International Journal of Computer Applications 118 (13):14-21, May 2015.

[24] Ravi (Ravinder) Prakash G, Kiran M." Characterization of Randomized External Source Output Map Reduce Expressions". International Journal of Computer Applications 123(14):9-16, August 2015.

[25] Ravi (Ravinder) Prakash G, Kiran M., Does there Exist Pruning Decomposition for MapReduce Expressions Arrangements?. International Journal of Computer Applications 125(12): 41-48, September 2015.

[26] Ravi (Ravinder) Prakash G, Kiran M: Can one find External Source Input Expressions for which there exist Map Reduce Configurations? International Journal of Computer Applications 128(12): 14-21, October 2015.